Exhibit 2

[redacted]

Preliminary UMP Viewer Application Functional Specification & Software Requirements

Section Three
Infrastructure to support Deployment of the UMP player

[redacted]

[redacted]

Executive Summary

NOTE: This is an executive summary of the overall document (all 3 sections). For convenience, during the preliminary definition process it is being repeated in each section.

This document describes the specifications, functionality and infrastructure of the UMG player. The document has three sections each addressing a part of the player and its operation. This document does not attempt to provide a fully comprehensive detailed specification of the player or its operation, but rather to provide a framework in which the player can be developed.

The three sections of the document are:

- Section One Architecture
- Section Two Functions
- Section Three Infrastructure

It is anticipated this document will be used to refine the design and will be iterated in line with the development of the player. It is intended that this development itself be iterative in nature and will have the following phases:

- Phase One Non functional GUI prototypes
- Phase Two Non functional prototype including iterated GUI components
- Phase Three Partially functional prototype with feature complete GUI and specification
- Phase Four Feature complete player in Alpha state
- Phase Five Feature complete pre production player in Beta State
- Phase Six Feature complete production player for General Release

This section of the document describes the major components of the systems that will be needed to support the deployment of the UMP. The document describes both those requirements for the anticipated trials and establishing these as a base for the further evolution of the and deployment of the player into a commercial delivery system.

[redacted]

Revision and Iteration History

Version	Publication date	Authors	Summary of Changes and updates
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]

Version note: All intervening versions numbers are internal revisions of the author and not for circulation.

[redacted]

Contents

[redacted]

Introduction

[redacted]

[redacted]

Assumptions

We are considering four possible scenarios which can affect the business and technical infrastructure:

Scenario 1 - All the objects are pre-packaged into secure containers (DigiBoxes) during the production process, prior to distribution. This includes both the "content and rules objects" coming from the distributors and "retail rules objects" coming from the retailers. The "content and rules objects" and the "retail rules objects" are kept in separate containers. When an order is placed based on a retail offer, we take the appropriate retailer's container and the object container(s) it controls and send them to the user. Unless there is a "magic" way to bind those containers together (which we are investigating), the "retail container" is likely to disappear during superdistribution.

Scenario 2 - All the objects, both the "content and rules objects" coming from the distributors and "retail rules objects" coming from the retailers, are not contained (i.e., not placed in DigiBoxes) while in storage. When a request comes in, a container (DigiBox) including content, owner and retail rules is created "on the fly". In this scenario the retailer's rights will persist in superdistribution.

Scenario 3 - All the objects are pre-packaged into secure containers (DigiBoxes) during the production process, prior to distribution. Unlike the Scenario 1, the content and all the rules (owner's and retailer's) controlling it are placed in the same container. When a request comes in, a prepared container (DigiBox) including content, owner and retail rules is sent. In this scenario the retailer's rights will persist in superdistribution.

Scenario 4 – A combination of 1 and 2. All the objects are pre-packaged into secure containers (DigiBoxes) during the production process, prior to distribution. When a request comes in, a new container of multiple DigiBoxes is created "on the fly". In this scenario the retailer's rights will persist in superdistribution.

To reduce the computational load in Scenarios 2 and 4, the containers for the most popular requests can be cached.

Each of the scenarios has its pluses and minuses and involves a number of business and technical trade-offs. We will be investigating with [redacted] what is the magnitude of these trade-offs.

For the purposes of this document we assumed that Scenario 1 is adopted. If a different scenario is accepted, it will necessitate changes to the architecture.

There are a number of other assumptions in this document which were not brought out here because they are highly contextual. Generally, the assumptions, issues, alternatives, questions, etc., are italicized.

[redacted]

Some Definitions

Note: This section will be a separate document and will be moved out of the later versions of this one. This section is here to establish some definitions which would be useful in the discussion that follows.

Container or Digibox (used interchangeably)

Always contains one or more Control Sets (= Business Rules) governing applicable objects. Usually will contain one or more Objects (or Properties).

Each Container (Digibox) will have a unique identifier – Container ID or CID - that will be a part of its reference, or URL. The Containers "in circulation" which have exactly the same content (Objects and Control Sets) will generally have the same CID, but Containers with different content will always have different CID's. The CID will be in the form of an alpha numeric string with a structured format. This might be in the form of the IEN 128 bar code standard or another standard numbering system. The CID may also have a Date/Time stamp and a version number, e.g. /emi.beatles.12345.v411.

There will be two types of Containers in the EMD: those that are produced by the distributors (they will usually have Objects in them) and those that are produced by the retailers and usually contain Control Sets only. *This is based on Scenario 1 of the Assumptions.*

The CID format should be selected carefully to meet the following goals: 1) it must allow to uniquely identify the Container, 2) it should be consistent with the identifications currently in use by the industry and UMG, 3) it should hopefully be descriptive, 4) it should be of reasonable length so that a person can actually type it, and 5) it should be automatically handled by the application with the least intelligence possible, similar to the way a router will handle an IP address.

Note that CID does not provide a detailed description of the content. Such a description will be provided in the container's table of contents or TOC.

Digital Objects or Content Objects

Any digital asset which UMG wants to store – an album, a song, text, graphics. If it is desirable to uniquely identify an object, this object will be assigned an Object ID or OID. An object can itself consist of multiple objects (i.e., the "Composite Object"). For example, an album consisting of 10 songs is an object in itself and will have its own OID. Each of the 10 songs is most likely an object and will also have its own OID. Additionally, some songs may have video and/or graphics associated with them (to be displayed on the screen), which in turn may have their own OID's.

Each OID is unique and will be a part of the object's reference, or URL. The identical Objects "in circulation" will generally have the same OID, but different Objects will always have different OID's. The OID will be in the form of an alpha numeric string with a structured format.

Some possible goals for the OID format: 1) allow to uniquely identify the Object, 2) be consistent with the current/emerging digital media asset identifications, 3) be descriptive, 4) be of reasonable length so that a person can actually type it, and 5) it should be automatically handled by the application with the least intelligence possible, similar to the way a router will handle an IP address.

The Composite Object's OID does not consist of the OID's of the objects inside it. Instead, from the player's perspective the Composite Object's OID is a server and the OID's of the objects inside it are the pages on that server. For example, /emi.phantom.1987.hifid may be the OID of the complete recording of "Phantom of the Opera" made by EMI in 1987 and encoded with high fidelity, while /emi.phantom.1987.hifid/angel_of_music may be the OID of the "Angel of Music" song from that recording. These content objects however may not be individually accessible or may require other objects

[redacted]

to be played at the same time. This constraint on accessibility will be implemented through the EDL/Associations function of the player. The associations will enforce the obligation to always keep material together or to keep any relationship the creator implies.

Alternatively, we can use Universal's SKU's as OID's. This is a very simple and short format. Its possible drawbacks: 1) it's non-descriptive, and 2) some digital objects we'd like to manage may not have SKU's.

Objects will have attributes. These will effectively be descriptive fields, thought they may be referenced within the association structure to create relationships between objects. The current set of properties fall into two categories, creator and owner. The creator set will include the Artist(s), Writer(s),Musicians(s), producer(S) and any other material that will assist in both finding the material in a search through the player Find function and provide the information for PRO and other mandated rights management functions. The Owner properties will define the owner of the copyright in both the song , Publisher(s), and the recording, Music Company(s) as well as the distribution channel (s) if appropriate. This data set will be used for the management of the rights of the content as well as the use in the Player Find functions. It is anticipated that the actual Audio and Video material will be watermarked with a set of this information in the form of an ISRC or ISWC code and that the player will likely check this status to validate the content and protect the IPR of the owners.

Channel Identifiers

The channel identifiers will provide the details of how the particular product offer came to market. This will include two data sets, the identifier of the Retailer (Retailer ID, or RID) and the Identifier of the Distributor (Distributor ID, or DID). These fields will be used as parameters when the object is referenced to redirect the reference to the correct site. These will also be used as the parameters in the usage and settlement functions. These ID's should be alphanumeric, simple and descriptive, e.g., DID = Universal, RID = tower.ny (Tower Records, NY server).

Digital Object Handle

Digital Object Handle (or DOH) is a URL which represents a request for a product. *Examples of handles below are given for illustration only!*

Suppose the EMD system is being used by a number of distributors and a user in LA logs onto Tower Records' website and selects an offer for Sinatra's "Everything Happens to Me" by Reprise Records. The DOH for this request will have a format /DID/OID/CID/RID, or as an example

/tw/reprise.sinatra.everything_happens_to_me.1996.hifid/ tower.sinatra.everything.123098/tower.la.doh

Note that the CID in this example is actually the one from the retailer.

In another example, a user gets Sinatra's "Everything Happens to Me" container superdistributed from a friend and tries to buy it. However, the Control Set does not allow purchasing of that particular container. Instead, the request gets sent to the EMD with a DOH = /DID/OID/CID:

/tw/reprise.sinatra.everything_happens_to_me.1996.hifid/reprise.sinatra.everything.101798.v.1doh

Note that the CID in this example is the one from the distributor.

What if we use an "SKU approach" to deriving ID's? Possible DOH's from a retailer and from superdistribution would look as follows:

/tw/audio.1234567/tower.6789.99/tower.la.doh

/tw/ audio.1234567/tw.987654.doh

[redacted]

High-Level Conceptual Architecture

The architecture should be designed around four main EMD functions:

- manage the digital assets
- securely deliver the content
- find the content
- track usage and provide settlements

Diagram 1 illustrates the proposed high-level architecture of the EMD service. The architecture consists of a number of blocks (components). Each of the architectural components is defined to perform a specific set of related functions while communicating with other components through specified interfaces. Once the interfaces are determined, the overall development can be broken down into a more manageable parallel development of individual systems which will be integrated together. Within this overall architecture the individual systems are outlined below.

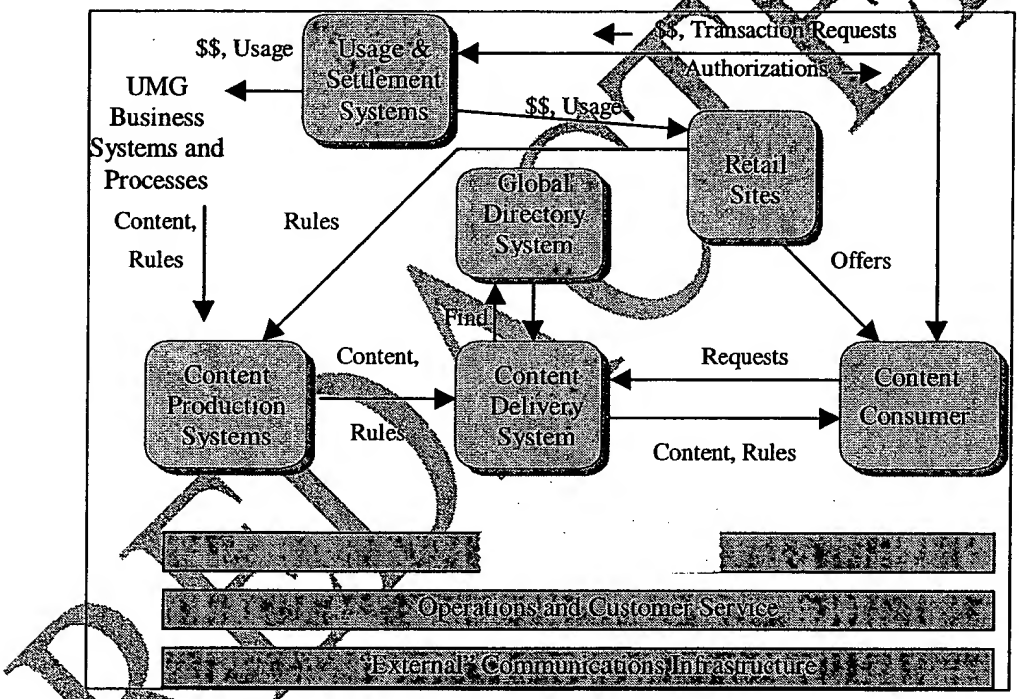


Diagram 1. High-Level Conceptual Architecture

Content Production System(s)

These systems provide processing, labeling and packaging of the content and associated business rules into the formats suitable for storage and distribution on the Internet. Content Production will be integrated into the existing business processes and systems. Within the EMD the Content Production will interface to the Content Delivery System to which it will forward the secured content and business rules (in Digiboxes) plus the information for finding the objects. The Content Production System (CPS) will also have an interface to the Retail Sites for providing information about the content availability and receiving associated retail business rules (which will be contained in DigiBoxes). *In the short-term the Content*

[redacted]

Production System will also interface to the Usage and Settlement Systems (USS) for payments and usage information; eventually, the USS interface will be migrated directly into the internal Business Operations Support Systems (BOSS).

Content Delivery System

This system is central to the architecture as it interfaces with the users and with most other systems of the service. It receives and stores secured content and rules from the Content Production System. It also receives content requests from the users, resolves these requests and delivers secured content to the users. Lastly, in some cases it accesses the Global Directory System to find the content (e.g., to support the player's Find function).
We may consider making users interact with the Global Directory System directly when conducting "Find" searches. This would be less objectionable to other distributors. My feeling right now is to have users go through the Delivery System, to have the default Content Delivery System set to UMG but allow users to change the default.

Global Directory System

The Global Directory System enables the Find function of the player. The Content Delivery elements will be optimized for quickly finding and delivering the content based on an easy-to-resolve, standard references. The Global Directory System will be designed to provide searches on many different indexes. The other function of the Global Directory System is to make the EMD service global by providing a "bridge" between different distributors. The production and delivery systems are likely to be internal to distributors (at least those who can afford them). Creating a Global Directory System which is "open" to all the content providers will make the overall EMD system more attractive to both users and distributors. *This system may have to be managed and maintained outside of EMG.*

Retail Sites

This will be the first point of contact with the user, where they have located a web server and downloaded the page containing the reference to the content (i.e., the DOH). All actions upon selecting this reference will then be transparent to the Retail Sites. The Retail Sites will include the links to the [redacted] application to download an Inter Rights Point (IRP) if they have not already done so. The Retail sites will also interface with the Content Production System to receive information about the content availability and provide associated secured retail business rules.

Content Consumer

Primarily the UMP player on a suitable host, initially the Wintel PC. This is described in detail in Section One of this document set. From the infrastructure perspective, the player needs to be supplemented with a high-speed communications device.

Usage and Settlement Systems

These systems perform mostly a clearinghouse function. They will provide the mechanism for the user to access the secured content by transacting for the appropriate rights. They will collect information on usage and transactions, which may include surveys and other non-financial information sets. The Usage and Settlement Systems interface with the users, with retail sites and with the internal business systems of content providers. The Usage and Settlement Systems should be coordinated with UMG's business systems. However, it must be remembered that in order for the EMD service to be "open" the Usage and Settlement Systems should be a "common resource".

[redacted] Application

Secures the content and business rules by providing encrypted containers; enforces the business rules.

[redacted]

Operations and Customer Service

Operational support for players in the field.

"External" Communications Infrastructure

Communications elements outside of our control which impact performance of the EMD service.

[redacted]

Content Production System

The Content Production System (CPS) will fulfill a number of important functions:

- create digital objects for the electronic delivery
- enable identification and tracking of digital content
- enable creation of the retail business rules
- secure the content together with the applicable business rules
- process payments and usage information (in the short term)
- interface with the "traditional" business processes and systems

It is particularly important to carefully lay out the long- and short-term strategies for the CPS and the migration path between the two, because the CPS is the system most affected by the "traditional" operations. The table below illustrates some of the possible differences between short- and long-term CPS implementations.

	Short-Term Implementation	Long-Term Implementation
Number of CPS's	One per distributor	Possibly one per label
Electronic Product Planning	Separate item	Integrated into the overall product plan
Asset Management	Local to CPS organization	Across the enterprise
Payment and Usage Information	Received by CPS	Received by business and marketing [redacted] systems
Interfaces to Other Systems and Organizations	E-mail, letter, fax, phone, some electronic data transfer	Transparent electronic data transfer

Table 1: Some Features of the Short-Term vs. the Long-Term CPS Implementation

Goals and Requirements

This is a partial list of the requirements and capabilities that CPS has to meet:

- accept the source content objects in different formats (CD, 128 Kbps, 384 Kbps, GIF, ...)
- accept attributes, properties and descriptions for content objects and maintain these associations throughout the production process, including associations with external (to CPS) objects
- encode the digital content object into different formats (128 Kbps, 384 Kbps, ...)
- provide transformation between different digital object formats
- register, index and store all digital content objects
  - automatically assign unique object ID's
  - index objects and query them without user's intervention, automatically extract attributes and create an annotation
  - create metadata "proxy" of an object
  - arbitrarily group digital objects into any classification hierarchy without duplication
  - support user-defined fields
- provide object access and searching capabilities
  - easy access via a user-friendly interface, including browsers
  - access to object's metadata
  - fast retrieval of large objects
  - index-based searching
  - content-based searching
  - attribute-based searching

[redacted]

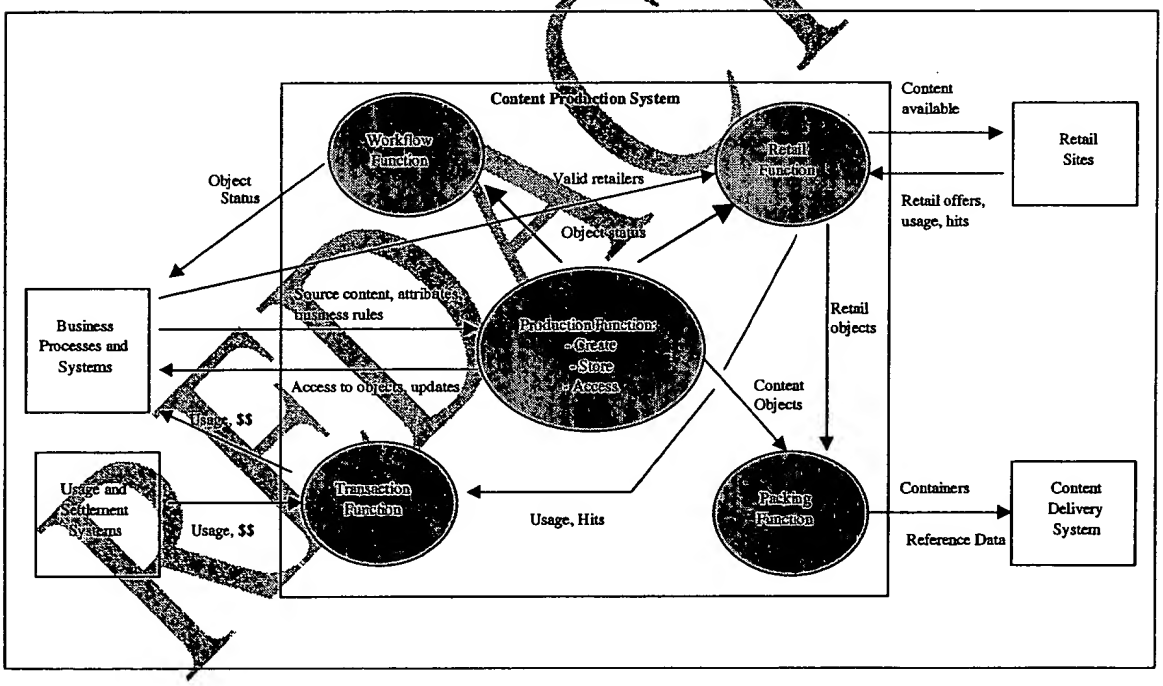
- owner- and usage-infromation searching
- track the status of a digital object throughout the production workflow process, provide audit trails
- support user- and group-level security
- provide cross-platform support (Win95, WinNT, MAC, Unix)
- store up to [50,000] objects initially, [1,000,000] objects in the long-term
- provide storage capacity of [4 TB] initially, up to [80 TB] in the long-term

Current Business Processes and Systems

Provide high-level functional description of the current UMG processes
....

Functional Implementation

The following diagram illustrates suggested high-level functional implementation of the Content Production System and its interfaces.



Daigram 2: CPS Functions

The CPS is divided into five functions: Production, Transaction, Retail, Packing, and Workflow. The division is somewhat arbitrary and can be changed to reflect the actual functionality of the development and implementation applications, as long as the overall scope of functions remains the same. The Production Function is much "larger" than the other four. However, it is anticipated that most of the Production Function requirements will be met by customizing an "off-the-shelf" media management package [redacted]

Production Function

What it does

This function is responsible for:

- Creating a digital object
  - Processing the source content
  - Processing content's properties, attributes and business rules
- QA'ing source and digital objects
- Indexing digital objects
  - Assigning object ID
  - Creating associations and classifications for searching
- Searching objects
- Accessing stored objects

Internal Interfaces

Internally, the Production function will have the following interfaces:

Packing Function - the Production function will provide to the Packing Function the digital content objects for securing into containers.

Retail Function - the Production function will provide to the Retail Function status of the digital content objects for communicating to the retailers, namely whether and when the object becomes available for electronic distribution.

Workflow Function - the Production function will provide to the Workflow Function status of the digital content objects for communicating internally, for generating work orders and providing information on when the object becomes available for electronic distribution.

All the internal communication will be implemented electronically and completely automated.

External Interfaces

Externally, the Production Function will interface with the Business Processes and Systems. The Production Function will receive the content objects in the source format, business rules associated with electronic distribution of these objects (release date, required associations, list prices, fidelities, etc.), and properties and attributes of these objects (artists, descriptions, text of songs, internal SKU, graphics, etc.). The Production Function will provide Business Processes and Systems with access to digital content objects.

It is anticipated that initially at least some portions of this interface will be manual.

Transaction Function

What it does

This function is responsible for collecting payment, usage, feedback, "hits" and other market data from the Usage and Settlement System and from the Retail Sites. *It is anticipated that this function is placed in the CPS temporarily, until it can be migrated to the Business Processes and Systems.*

Internal Interfaces

[redacted]

Internally, the Transaction function will have the following interfaces:

Retail Function - the Transaction function will receive from the Retail Function information about usage, hits, comments, etc. collected from the retailers.

All the internal communication will be implemented electronically and completely automated.

External Interfaces

Externally, the Transaction Function will interface with the Business Processes and Systems and with the Usage and Settlement Systems. The Transaction Function will receive from the Usage and Settlement Systems the usage and payments information. The Transaction Function will provide Business Processes and Systems with the payment information and the usage and other market data aggregated from the USS and the Retail Sites.

It is anticipated that initially at least some portions of this interface will be manual.

Retail Function

What it does

This function is responsible for communicating with the Retail Sites. It will inform the retailers of content's availability, allow them to create retail "objects" (i.e., offers which can be processed into containers), keep track of the status of retailers' offers, and ensure that only valid retailers can access the system and only with valid offers.

Internal Interfaces

Internally, the Retail function will have the following interfaces:

Production Function - the Retail function will receive from the Production Function status of the digital content objects for communicating to the retailers, namely whether and when the object becomes available for electronic distribution.

Transaction Function - the Retail function will send to the Transaction Function information about usage, hits, comments, etc. collected from the retailers.

Packing Function - the Retail function will send to the Packing Function "retail business rules" objects for inclusion in containers.

All the internal communication will be implemented electronically and completely automated.

External Interfaces

Externally, the Retail function will interface with the Retail Sites and the Business Processes and Systems.

The Retail Function will allow retailers to log on via their browsers, check the information about the content available for sale over the Internet and "fill out" a form creating a "retail business rules" object. The Retail Function will also actively "push" out to the retailers information about the new content and reminders of the timing of their retail offers (e.g., if an offer will expire soon).

The Retail Function will receive from Business Processes and Systems information about the Retail Sites which can access the system.

[redacted]

It is anticipated that initially at least some portions of this interface will be manual.

Workflow Function

What it does

This operational function is responsible for keeping track of the "work-in-progress" objects and for creating "work orders" to "pace" the work.

Internal Interfaces

Internally, the Workflow function will have the following interface:

Production Function - the Workflow function will receive from the Production Function status of the digital content objects and the associated data (e.g., the source content has been received and encoded but one of the files required by "business rules" associations is missing) until the object becomes available for electronic distribution.

All the internal communication will be implemented electronically and completely automated.

External Interfaces

Externally, the Workflow function will interface with the Business Processes and Systems and with the CPS's personnel. The Workflow Function will communicate to the Business Processes and Systems information about the status of the "work-in-progress" objects and the items remaining to complete the work.

It is anticipated that initially at least some portions of this interface will be manual.

Packing Function

What it does

This function is responsible for the construction of the secure containers, with the inclusion of the actual content (Audio, Video, Graphics etc) and the associated business rules as applicable. It is also responsible for transferring the containers and the reference data to the Content Delivery System. The Packing Function will be derived from the [redacted] packer application.

Internal Interfaces

Internally, the Packing Function will have the following interfaces:

Production Function - the Packing Function will receive from the Production Function the digital content objects prepared for the electronic distribution, including the content and the associated business rules.

Retail Function - the Packing Function will receive from the Retail Function the "retail business rules" objects prepared for the electronic distribution.

All the internal communication will be implemented electronically and completely automated.

External Interfaces

Externally, the Retail Function will interface with the Content Delivery System. The Packing Function will send to the Content Delivery System secured containers (DigiBoxes) prepared for the Internet distribution,

[redacted]

together with the information about the "contained content" for resolving customers' requests and for inclusion in the Global Directory (e.g., to support the Find function).

It is anticipated that this interface will be electronic and completely automated.

Content Production System Architecture Overview

The following diagram illustrates a possible CPS system architecture.

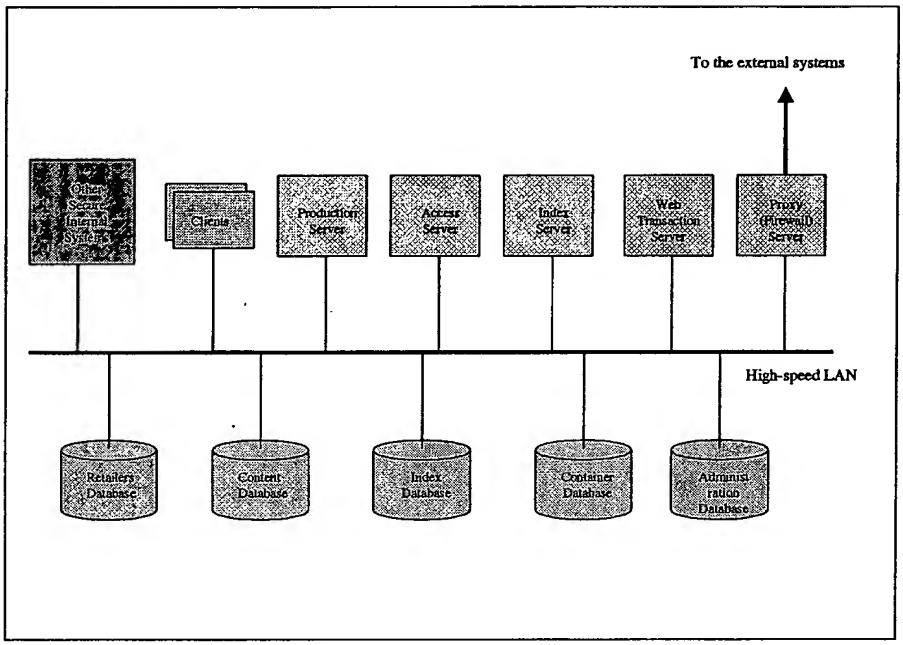


Diagram 7 – Possible Content Production System Architecture

It consists of the following elements:

Production Server - manages Production Function "front-end", Packing and Workflow Functions.

Index Server - manages "metadata" storage and access of the Production Function. May not be necessary in the short-term, while the content database is small.

Access Server - manages actual content storage and access of the Production Function. May perform the Index Server functions in the short-term, while the content database is small.

Web Transaction Server - manages Retail and Transaction Functions.

Proxy Server - provides a secure firewall for the content production.

Content Database - stores the actual digital objects together with their properties, attributes, descriptions and associations. The key index is OID.

Index Database - stores information about digital objects - their key properties, attributes, descriptions and associations. The key index is OID; highly searchable.

Container Database - stores the secure containers (DigiBoxes). Can be searched on both CID and OID indexes.

Retailers Database - stores the retailers information and their "retail offer" objects. Can be searched on both RID and OID indexes and other information.

Administration Database - stores the information about payments, usage and other market data. Can be searched on RID, OID, CID and other indexes.

Recommended Solution

...

[redacted]

Content Delivery System

This system will provide the services, functions and performance required to support the delivery of the content to the consumer. The Content Delivery System (CDS) will fulfill the following functions:

- receive and store secure containers and associated information
- quickly find and retrieve the requested content
- establish communication and transfer the requested content without customer's intervention, completely and without errors
- provide customers with alternatives when the requested content is not available
- operate in an on-line environment with up to [50,000] requests/day short-term, [5,000,000] requests/day long-term.

Functional Implementation

The following diagram illustrates suggested high-level functional implementation of the Content Delivery System and its interfaces:

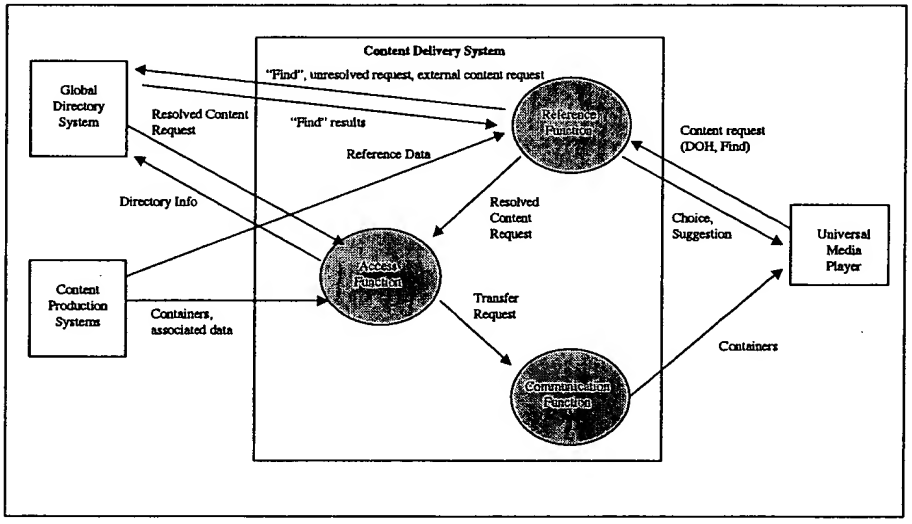


Diagram 4: Content Delivery System Functional Diagram

The CDS is divided into three functions: Reference, Access, and Communication. The division is somewhat arbitrary and can be changed to reflect the actual functionality of the development and implementation applications, as long as the overall scope of functions remains the same (probably, a more granular approach – i.e., a larger number of "smaller" functions – will be needed).

Reference Function

<u>What it does</u>

[redacted]

This function is always the one receiving the customer's request. The request would be either for an object defined by its DOH or a "Find" request for search based on certain keywords or object attributes. The DOH can be a "handle" from a retailer's site pointing to an offer in the form of /DID/OID/CID/RID, or a "handle" in some other form, e.g., an OID typed in by the user *(we have to dig in deeper on what kinds of "handles" are possible / allowed).*

There is only one Reference Function per distributorship (in the long-term the actual implementation might be distributed for faster response). A request would go to the Reference Function of a particular distributorship because either 1) the DID in the handle is for that distributorship, or 2) this is a "Find" request and the distributorship is the default one for that player.

The Reference Function is responsible for resolving the request and redirecting it if necessary. In order to do that the Reference Function keeps track on a distributorship-wide basis of all the objects (both content and retail offers) and containers authorized by the distributorship for electronic distribution. Specifically, the Reference Function maintains information about all the OID's, CID's and RID's, their relationships (associations) and selected relevant information (e.g., the expiration date, a brief description of offers and content objects, etc.). Because "handles" can remain in circulation long after the applicable offer has expired, the Reference Function should maintain the information for at least [6?] months after the offer has expired.

If the DOH request is successfully resolved (i.e., there is one and only one object or combination of objects corresponding to this "handle" and this object(s) is available at the distributorship), the resolved request is forwarded to the Access Function. *Note that this is somewhat different from the 5/27 version of this document, where all the requests had to go through a secure X.500 directory service. Here if the request is for the object(s) from the same distributorship as the Reference Function, it can stay within the distributorship's network where the communication can be secured by other (besides the X.500 service) means. This assumes that either there is a single Access Function per distributorship, or the Reference Function knows which Access Function to go to. The Reference Function has an option to go through the X.500 directory for all requests, and some smaller distributors may choose to implement it this way.*

If the DOH request can not be successfully resolved because, for example, this particular offer has expired but there are objects or combination of objects which can be substituted for this "handle" and these object(s) are available at the distributorship, the Reference Function will offer to the customer a menu of suggestions. Business rules for creating these suggestions will have to be defined, e.g., first offer(s) for the same content object (OID) from the same retailer (RID), then offer(s) for the same content object (OID) from other sources (e.g., other retailers or the distributor itself). If the same OID can't be found in the Reference Function's database, the Reference Function will forward the request to the Global Directory Service to see if the same object can be found elsewhere. If it is found, the customer will be offered it. If not, the Reference Function will offer to the customer to conduct a more detailed search (it is assumed that the Reference Function will not offer to substitute a different object (OID)).

If the request is for a "Find" operation, the Reference Function will redirect it to the Global Directory System. *Alternatively, the Reference Function may first try to search within the distributorship, by passing the request to its Access Function. This preferential treatment may, however, have a negative effect on other distributors / independents joining the service (a good example is the airline reservations industry where multiple systems ended up competing, each favoring some of the airlines).*

There are a number of possible implementations of how the Reference Function will interact with the Global Directory Service (GDS):

1) *Once the Reference Function redirects the request to the GDS, the customer establishes a session directly with the GDS and the Reference Function is not involved. When the customer finds the object he/she wants, the GDS will redirect the customer to (needs to be resolved) either a) the Reference Function of the distributor of the object, or b) the Access Function controlling the object (which may or may not be UMG's), or*

[redacted]

2) *The Reference Function queries the GDS in the background, i.e., from the customer's perspective he/she is still interacting with the same Reference Function. When the customer finds the object he/she wants, the Reference Function will redirect the customer to the Access Function controlling the object. If the Access Function is not UMG's, the redirection will be via the GDS.*

In the following discussion we assumed that Approach 2 is used.

Internal Interfaces

Internally, the Reference Function will have the following interface:

Access Function - the Reference Function will send to the Access Function the resolved content request. The request will include the OID, CID and RID information plus the user data (e.g., TCP/IP address). *Which of these ID's are necessary and which are redundant depends on how the objects are stored (in containers or not) and the containers formed (in advance or on-the-fly). Depending on how the issues above are addressed, the Reference Function may also send search requests to the Access Function and/or the Access Function may participate in populating the Reference Function's database (if there are multiple Access Functions per distributorship and if the GDS is bypassed in intra-distributorship requests).*

All the internal communication will be implemented electronically and completely automated.

External Interfaces

Externally, the Reference Function will interface with the customer's UMP, with the Global Directory System and with the Content Production System. As described above, from the UMP the Reference Function will receive content requests. The Reference Function will send queries to the GDS based on the "Find" requests and unresolved requests. From the CPS the Reference Function will receive the "reference data" which will include all the OID's, CID's and RID's available for electronic distribution from this distributor *(as discussed above, if there is more than one Access Function per distributorship, the "reference data" may have to come from the Access Functions).*

It is anticipated that all the external communication will be implemented electronically and completely automated.

Access Function

What it does:

This function is responsible for storing, tracking and accessing the objects/containers received from the Content Production System. It keeps detailed information about the available content (allowing searches on many different indexes and keywords) and supplies this information (probably in a summary form) to the Global Directory System (GDS).

The primary operational activity of the Access Function is to receive a request for content from the Reference Function, determine where the content is physically available and pass the transfer request to the Communication Function. *The exact division of responsibilities between the Access Function and the Communication Function will need to be discussed, particularly in the area managing the performance – scheduling transmission, balancing loads, selecting the optimum location for downloading. For the sake of argument we will assume that these responsibilities reside in the Communication Function and that the Access Function will determine where the content is available and will send the content's ID's and locations to the Communication Function.*

[redacted]

We assume that there will be only one Access Function per distributor, albeit existing in multiple instances. If this is not correct (e.g., each label implements its own Access Function), it will affect how the Reference Function obtains its "reference data".

Internal Interfaces

Internally, the Access function will have the following interfaces:

Reference Function - the Access Function will receive from the Reference Function the resolved content request. The request will include the OID, CID and RID information, plus the user data (e.g., TCP/IP address).

Communication Function - the Access function will send to the Communication Function the data transfer request, consisting of the ID's of the data to be transferred, the locations where the data is stored and the user's address. *The exact nature of the ID's needed by the Communication Function will depend on how the objects are stored and processed into containers. If, as assumed here, the objects are placed into downloadable containers in the CPS, the Access Function needs to communicate the CID(s) only. If, on the other hand, CDS needs to form containers out of the objects prior to downloading, the Communication Function will contain a "Packing Function" and the OID's will be communicated.*

All the internal communication will be implemented electronically and completely automated.

External Interfaces

Externally, the Access Function will interface with the Global Directory System and with the Content Production System. The Access Function will send "directory information" to the GDS about the content objects being controlled by the Access Function. The "directory information" will include OID plus a condensed version of the object's properties and attributes (e.g., title, artists, musicians, composer, producer, writer, publisher, distributor, date of performance, fidelity, keywords, etc.) to assist the Find function. The retailers information will not be included in the GDS. The Access Function may (depending on how some of the assumptions above are resolved) receive content requests and queries from the GDS redirected from "other" Reference Servers.

From the CPS the Access Function will receive the containers or objects and the associated data which will include all the OID's, CID's and RID's available for electronic distribution from this distributor plus all the "directory information".

It is anticipated that all the external communication will be implemented electronically and completely automated.

Communication Function

What it does

This function is responsible for managing all the downloading of content from the CDS to the consumers. In interacting with a customer, the Communication Function has to be able to download large objects while guaranteeing completeness and freedom from errors (i.e., after the download the object at the UMP should be exactly the same as the object at the CDS) even over an unreliable communications path. It should be user-friendly, e.g., require no user intervention while providing useful information, such as storage required and time left. It should be able to multitask with other resources on the UMP's platform.

The other critical aspect of the Communication Function is its ability to handle multiple simultaneous requests. It should be able to optimize / balance the load across multiple servers and datastores. However,

[redacted]

this by itself won't be enough. It is anticipated that in the EMD service a small portion of the content will generate most of the demand, and that "hits" or promotions will create temporary but very significant "spikes" in user requests. A system capable of handling such input "spikes" without some way of "smoothing" the output is likely to be prohibitively expensive or even impossible (due to reliance on the "external" telecommunications infrastructure). The Communication Function should be intelligent enough to deal with such situations via scheduling and multicasting. The multicasting capability will exploit the fact that the demand "spikes" will most likely be caused by requests for the same content. Instead of sending the file to each customer individually, the file can be multicast to a group of customers. For the multicasting to be effective, it would require a sophisticated scheduling mechanism to form these multicasting groups in real time. Scheduling mechanism can further ease the load by offering downloads during off-peak hours (e.g., in the middle of the night). The performance can be further improved by a distributed architecture where the content is stored in multiple places so that it can be "pushed" to the end users from a local server. This in turn requires sophisticated content replication and synchronization techniques.

Most likely all of the above measures will be needed for deploying the EMD service on any kind of mass scale. Thus, the Communication Function will be a complex one, requiring careful design and sophisticated interaction of traditional communications protocols, server load balancing, scheduling, multicasting and database replication.

Internal Interfaces

Internally, the Communication Function will have the following interface:

Access Function - the Communication Function will receive from the Access Function the content transfer request. The request will include the ID's of the data to be transferred, the locations where the data is stored and the user's address. *The exact nature of the ID's needed by the Communication Function will depend on how the objects are stored and processed into containers (see the discussion above).*

All the internal communication will be implemented electronically and completely automated.

External Interfaces

Externally, the Communication Function will only interface with the customer via the UMP. The Communication Function will negotiate the transfer session, download the container(s) and verify completeness and correctness of the download.

It is anticipated that all the external communication will be implemented electronically and completely automated.

Content Delivery System Architecture Overview

The following diagram illustrates a possible CDS system architecture.

[redacted]

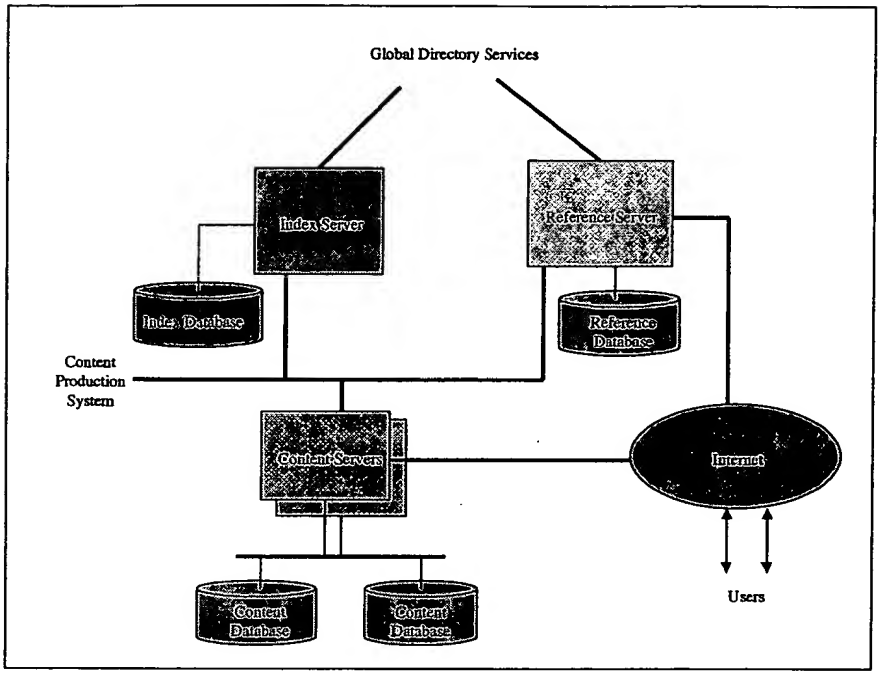


Diagram 5 – Possible Content Delivery System Architecture

It consists of the following elements:

Reference Server - manages the Reference Function. One per distributor, although is likely to eventually have multiple instances to improve customer service. Resolves customer requests and forwards them to the Index Server or the Global Directory Service.

Index Server - manages the "front-end" of the Access Function. May also host the "front end" of the Communication function (e.g., scheduling and load balancing). Keeps track of where the content is stored, all the related ID's, properties, attributes, "directory information". Allocates storage for the content being received from the CPS. Receives content requests from the Reference Server and translates them into the transfer requests for the Content Servers. Updates GDS. Assumed one per distributor, although is likely to eventually have multiple instances to improve customer service.

Content Servers - manage the "back-end" of the Access Function (actual content storage and retrieval) and the Communication Function (except that scheduling and load balancing may be centralized at the Index Server). If the containers are formed "on-the-fly" (rather than prepared in advance), performs the Packing Function (see CPS). Highly distributed. In each instance is a "server farm" interconnected with a very high-speed local network.

Content Database - stores the actual containers or digital objects, depending on what's received from the CPS. The key index is either CID or OID. Highly distributed.

Index Database - stores information about digital objects - their key properties, attributes, descriptions and associations. The key index is OID; highly searchable. Assumed one per distributor, although is likely to eventually have multiple instances to improve customer service.

[redacted]

Reference Database - stores information necessary to resolve DOH's, mainly OID, CID and RID. One per distributor, although is likely to eventually have multiple instances to improve customer service

Communication Links – high-speed dedicated connections to the Internet from the Reference Server and the Content Servers; *high-speed on-demand connections for the private network interconnecting the servers, CPS and GDS (as shown) or a virtual private network over the Internet – if security, reliability and throughput concerns can be addressed.*

Recommended Solution

...

[redacted]

Global Directory System

The Global Directory System (GDS) will serve a dual purpose:

1) to support the "Find" function in the UMP, and
2) to provide the interconnection point between Content Delivery Systems of different distributors compatible with the EMD service.

The "interconnection" concept is illustrated in the following diagram.

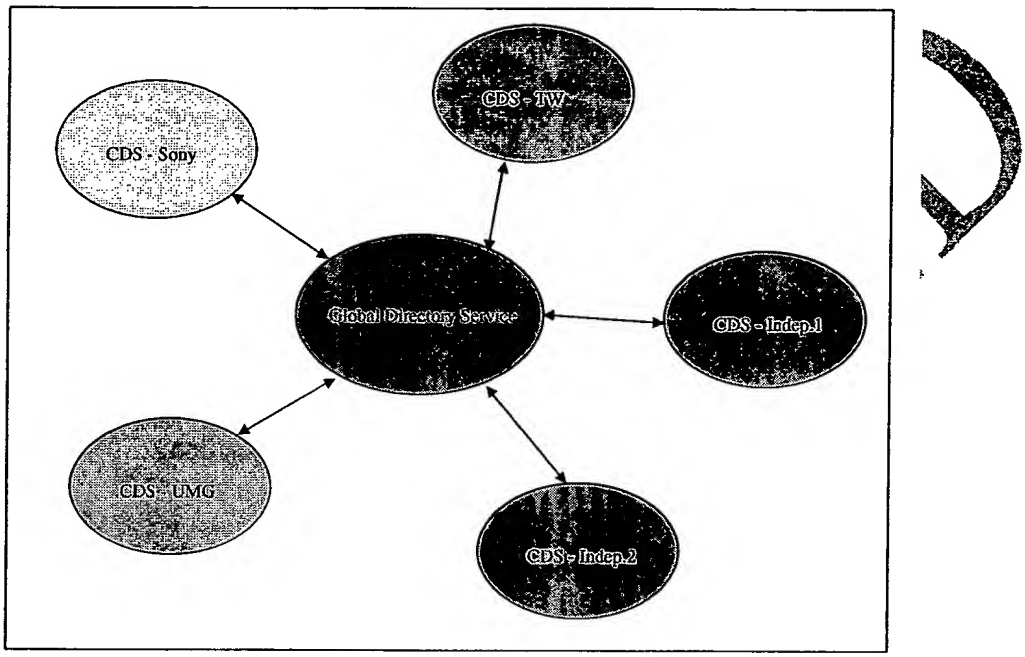


Diagram 6 – Global Directory System as an Industry Interconnection Point

The analogy is that of the Internet, where multiple service providers can connect at one of the defined interconnection points. The approach is to provide a single secure directory services function, such that all access to the content outside of a distributor's "private network" is undertaken through this transparent service. Thanks to the GDS there is a motivation on the part of other distributors and independents to become EMD-compatible: they can maintain their own production and delivery and still have access to all the UMP-equipped customers, or they can outsource their production and/or delivery functions (which can be quite expensive to implement) to UMG but maintain their brand before the customers.

Given that the GDS is defined as an industry-wide resource, it probably best be maintained by an industry organization, such as IFPI. If the consensus on GDS is not achieved in a reasonable time-frame, UMG's service launch won't be delayed – the "Find" function will be enabled in the Index Server of the Content Delivery System, which has all the information needed (however, the searches will be limited to the UMG material and those partners who chose to either include their information in the UMG's Index Database or to connect their Index Server to UMG's Reference Server).

It is envisioned that the GDS database will eventually have a reference to every digital content object available for electronic distribution in the EMD-compatible format. This reference should include a unique OID, the appropriate Index Server and at least those properties and attributes which are most likely to be used in a "Find" function search. Except for the Index Server, these information is created during the production process. Given that the standards for digital objects' identification are not well developed, it is

[redacted]

important to review the work of relevant organizations and vendors to ensure that EMD's formats are compatible with whatever *de juro* or *de facto* standards emerge.

Global Directory System Architecture Overview

The following diagram illustrates a possible CDS system architecture.

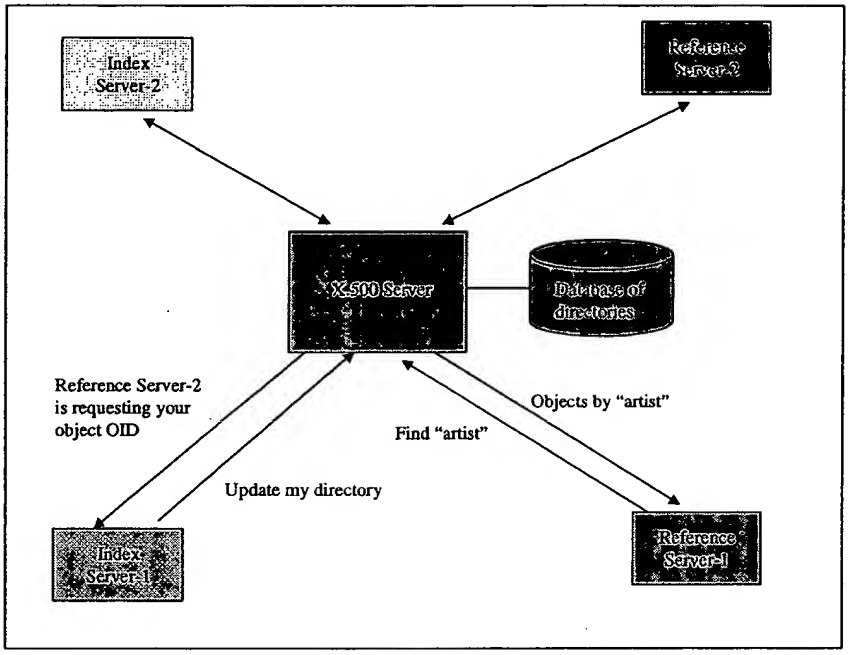


Diagram 7: Possible Global Directory Service Architecture

The envisaged approach is to use an X500 directory services model. The X500 directory server system will act as a gateway between the reference servers and the content databases and server farms where the content is stored (via appropriate index servers). The X500 system will only transact with legitimate reference servers and provide access to the content for supported connections. The X500 server will translate the request from the reference server into an instruction for the index server which "owns" the object to respond to the reference server and, eventually, for the correct database to send the content to the user.

Recommended Solution

...

Usage and Settlement Systems

Consist of Finance Clearing House, Information Clearing House, and the Deployment Manager.

The Finance Clearing House (FCH) will manage payment processing when purchasing occurs. When a customer wants to access the content within a Digibox, he/she will transact with the FCH to obtain the appropriate rights. FCH will have gateways to banks and credit card processors to handle various forms of payment. FCH should be able to aggregate multiple transactions into a single one. FCH has to maintain extensive audit records.

The deployment manager will provide the user with the [redacted] with which to undertake the transactions with the FCH and gain access to the content. This may become a specific reference to the X500 gateway with a separate server for this purpose, or may be a free standing entity

The Information Clearing House will collect usage and survey information and offer data analysis and reporting services.

In the long-term both FCH and ICH will have to be integrated with the Universal's Business Operations Support Systems (BOSS).

[redacted]

Retail Sites

Basically a web server. Once accepted into the EMD, becomes a part of UMG's extranet. Needs an application to log onto the Web Transaction Server of the Content Production System, to fill out a form creating a "retail business rules" object and incorporate the appropriate DOH's into its pages. May eventually be allowed to do the packing internally.

[redacted] Application

[redacted]

Support Functions

- UMP Applications
- Customer Support
- Development Support

Application support

Elements needed for player deployment and support of the delivery of music to the consumers. This area spans the processes and supporting systems and infrastructure to ensure that the content reaches the consumer in the manner intended

The player has a number if functions that require support.

Content (Digibox) delivery
Content referencing
Buddy list servers
Content related sites (band sites)

Mail Proxy

A standard Mail proxy server set up to support all the email addresses included in the Digibox's as content for contact with the artist, producer, label or for promotional purposes.

Chat- Buddy Gateway

This will be a server to either directly support a chat service contained within the Digibox or provide a gateway to another service such as AOL Instant Message. This gateway may include CGI type scripts to undertake additional parsing on the messages received to match the current obligations and requirements of the Artist or Label.

In addition this is the server where "live" Artist interaction will be channeled through.

This will also provide access to the simultaneous play functions to be offered to the users and as such may support a number of scripts and functions to support multiple users playing a particular piece of Artist material. This may involve the use of streaming and multicasting.

Domain Servers

The Domain servers will provide the prime addressing for the content. This is likely to be in the form of Domain.newmusic/content/artist/label. This will be the location of the web pages identified within the Digboxes.

[redacted]

Operational Support

In addition there will need to be systems to support the players operation in the field. This will initially involve the users and trial members but will expand to support users, developers and potentially other content providers.

This will also include the support of some of the financial transactions and their implications

Customer Support

After the player is deployed there are significant need for a customer support function. This will involve

Dispute resolution
Back and Restore
Product support
Customer service

A key element for providing this facility will be a customer facing call centre. This will provide the telephonic and electronic interface to consumers to provide the needed support functions.

The call centre will provide

Developer Support

Clearing House Integration

[redacted]

Infrastructure Deployment Plan

- Pilot
  - Functionality to be tested
  - Sizing
  - Implementation

Trial Size estimates

50 albums at release

- Limited Commercial Rollout
  - Functionality to be tested
  - Sizing
  - Implementation

- Full Commercial Rollout
  - Sizing
  - Implementation

[redacted]

Use Scenarios

- Main Scenaios
  - New Content Offer
  - Retailer Changes Business Rules
  - Content Provider Changes Business Rules
  - User Downloads UMP Client
  - User Downloads Content
  - User Purchases Content
  - Superdistribution
  - Settlement
- Secondary Scenarios and Exception Handling